REMARKS

In the patent application, claims 1-24 are pending. In the office action, all pending claims are rejected.

Applicant has amended claim 2 to change "the decoder" to "a decoder" as suggested by the Examiner. Applicant has also amended claim 16 to correct for a typographical error.

No new matter has been introduced.

At section 2 of the office action, claim 2 is objected to because of informalities. Applicant has amended claim 2 to overcome the objection.

At section 4, claims 1-5, 7-12, 15, 17 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Lee et al.* ("A very low bit rate speech coder based on a recognition/synthesis paradigm" IEEE Trans on Speech and Audio Processing, Vol. 9, No. 5, July 2001, hereafter referred to as *Lee*), in view of *Gao* (U.S. Patent No. 6,449,590).

In rejecting claims 1, 11, 17 and 20, the Examiner states that *Lee* discloses a method and system for improving coding efficiency having the following steps:

creating a plurality of simplified pitch contour segment candidates, each candidate corresponding to a sub-segment of the audio signal (Section V.A., pages 486-487);

measuring deviation between each of the simplified pitch contour segment candidates and the pitch values in the corresponding sub-segment; and

selecting a plurality of consecutive segment candidates to represent the audio segment (Section V.A., Pages 486-487; Figure 5); and

coding the pitch contour data in the sub-segments of the audio signal corresponding to the selected segment candidates (Section V. Page 486).

The Examiner admits that *Lee* fails to specifically suggest that the start and end points of a pitch contour sub-segment candidate may vary from that of the original speech sub-segment. The Examiner points to *Gao* for disclosing a means for time-warping the start and end points of a speech-sub-segment (col.2, line 17 to col. 43, line 14).

The Examiner states that it would be obvious for one skilled in the art to modify the approximation method used by *Lee* using the time-warping method in *Gao* in order to implement an efficient pitch contour coding process.

Applicant respectfully disagrees.

The speech coding process according to *Gao*

It is respectfully submitted that *Gao* discloses a method of speech processing wherein the encoder performs high-pass filtering and applies a perceptual weighting filter for providing weighted speech signal, and a pitch preprocessing operation is applied to warp the weighted speech signal in order to match the interpolated pitch values that will be generated by the decoder (col.5, lines 52 to 65). *Gao* uses high-pass filtering, perceptual weighting and speech signal warping to support lower bit-rate encoding modes. All these three steps are necessary to produce a linear pitch lag contour (see Figure 8c) from a non-linear pitch lag contour (Figure 8b). In fact, *Gao* discloses a method where the encoder generates a pitch lag contour by using estimates of a previous pitch lag and a current pitch lag of the speech signal and then warp the speech signal by temporally deforming the weighted speech signal in order to conform to the generated pitched lag contour (col.70, lines 47-53; col. 71, lines 10 – 17).

The speech coding process according to *Lee*

Lee discloses a method to substitute non-linear contour segments with linear contour segments. *Lee* simply picks a start point in the pitch contour and searches for an end point in the pitch contour that produces a linear segment having an error from the original contour segment smaller than d_{max}.

The differences between the approaches in *Lee* and *Gao*

Lee's coding method is contour-wise rather than frame-wise (last paragraph of left column on page 486). While *Gao* also discloses a method to substitute non-linear pitch lag contour segments with linear pitch lag contour segments, *Gao*'s coding method is frame-wise or subframe-wise (col. 42, lines 17 – 34). *Gao* uses high-pass filtering of the speech signal and a perceptual weighting filter for providing weighted speech signal. *Lee* does not use those steps.

There would be no motivation or incentive to combine the approaches in *Lee* and *Gao*

In order to raise a 103 rejection, the Examiner must show why a person skilled in the art would want to apply the method as disclosed in *Gao* to the method as disclosed in *Lee*. The Examiner fails to show such motivation.

lack of motivation

There are many reasons why there would be no motivation to combine the approaches in *Lee* and *Gao*.

A. Complexity

As admitted by the Examiner, *Lee* uses the following four steps for pitch contour coding:

1) creating a plurality of simplified pitch contour segment candidates, each candidate corresponding to a sub-segment of the audio signal (Section V.A., pages 486-487);

2) measuring deviation between each of the simplified pitch contour segment candidates and the pitch values in the corresponding sub-segment; and

3) selecting a plurality of consecutive segment candidates to represent the audio segment (Section V.A., Pages 486-487; Figure 5); and

4) coding the pitch contour data in the sub-segments of the audio signal corresponding to the selected segment candidates (Section V. Page 486).

Gao uses a different approach in pitch contour coding. *Gao's* process involves at least three steps (col.5, line 52 – 64):

a) high-pass filtering the speech signal;

b) applying a perceptual weighting filter to the high-pass filtered speech signal for providing weighted speech signal, and

c) warping the weighted speech signal in order to match the interpolated pitch values that will be generated by the decoder.

None of the steps in *Gao* are used in *Lee*. Thus, in order to combine the method as disclosed in *Gao* to the method as disclosed in *Lee*, one must use all of the seven steps as shown above. The combined method requires a very complex encoder.

B. Compatibility

As mentioned earlier, *Lee's* coding method is contour-wise rather than frame-wise, whereas *Gao*'s coding method is frame-wise or subframe-wise. Furthermore, in order to time-warp the pitch lag contour, *Gao* requires applying a perceptual weighting filter to provide weighed speech signal. It is uncertain whether *Lee* can use the weighed speech signal to create a plurality of simplified pitch contour segment candidates, each candidate corresponding to a sub-segment of the weighted speech signal, and then measure the deviation between each of the simplified pitch contour segment candidates and the pitch values in the corresponding sub-segment.

C. *Lee* alone can accomplish what the combination of *Gao* and *Lee* may provide

Gao uses a time-warping method to replace non-linear pitch lag contour segments with linear pitch lag contour segments. The objective is to lower the coding bit-rate so as to meet a certain encoding mode.

Lee alone can lower the coding bit rate to meet a certain encoding mode by changing d_{max}. For example, if a high bit-rate is available, *Lee* may use a smaller d_{max} to improve the encoding accuracy. But when a lower bit-rate is required, *Lee* can use a larger d_{max} in order to reduce the number of linear pitch contour segments. There is no need to introduce three additional steps as required in *Gao*.

D. *Gao's* approach is not beneficial to the present invention

The present invention can lower the coding bit-rate to meet a certain encoding mode by changing the predetermined error value in the comparison step **508** as shown in the flowchart **500** of Figure 4 (p.11, lines 17 - 24). There is no need to use the time warping techniques as disclosed in *Gao*.

<u>*Lee*, in view of *Gao*, fails to render the present invention obvious</u>

In sum, *Lee* does not require the approach as used in *Gao* in order to meet a certain bit-rate requirement. The present invention does not require the approach as used in *Gao* in order to meet a certain bit-rate requirement. *Lee* and *Gao* may not be compatible to each other. Even if they are compatible, the combination of *Lee* and *Gao* yields an unnecessary complex encoding

system. The Examiner fails to show why a person skilled in the art would choose such a complex encoding system when a much simpler encoding system can achieve the same result. lack of motivation

For the above reasons, it is respectfully submitted that *Lee*, in view of *Gao*, does not render the invention as claimed in claims 1, 11, 17 and 20 obvious.

As for claims 2-5, 7-10, 12 and 15, they are dependent from claims 1 and 11 and recite features not recited in claims 1 and 11. For reasons regarding claims 1 and 11 above, it is respectfully submitted that claims 2-5, 7-10, 12 and 15 are also distinguishable over the cited *Lee* and *Gao* references.

At section 5, claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Lee*, in view of *Gao* and further in view of *Swaminathan et al.* (U.S. Patent No. 5,704,000, hereafter referred to as *Swaminathan*).

The Examiner cites *Swaminathan* for disclosing a means for selecting from a plurality of pitch candidates corresponding to pitch parameters of a specific pitch period.

It is respectfully submitted that claim 6 is dependent from claim 1 and recites features not recited in claim 1. For reasons regarding claim 1 above, claim 6 is also distinguishable over the cited *Lee*, Gao and *Swaminathan* references.

At section 6, claims 13-14, 16, 18, 19 and 21-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Lee*, in view of *Gao* and further in view of *Lumelsky* (U.S. Patent No. 6,246,672).

The Examiner cites *Lumelsky* for disclosing a storage means for storing encoded audio data.

It is respectfully submitted that claims 13-14, 16, 18, 19 and 21-23 are dependent from claims 11, 17 and 20 and recites features not recited in claims 1, 11 and 20. For reasons regarding claims 1, 11 and 20 above, claims 13-14, 16, 18, 19 and 21-23 are also distinguishable over the cited *Lee*, *Gao* and *Lumelsky* references.

As for claim 24, it claims a communication network comprising a decoder as claimed in claim 17. For reasons regarding claim 17 above, it is respectfully submitted that claim 24 is also distinguishable over the cited *Lee*, *Gao* and *Lumelsky* references.

CONCLUSION

As amended, claims 1-24 are allowable. Early allowance of all pending claims is earnestly solicited.

Respectfully submitted,

Kenneth Q. Lao
Attorney for the Applicant
Registration No. 40,061

WARE, FRESSOLA, VAN DER SLUYS
& ADOLPHSON LLP
Bradford Green, Building Five
755 Main Street, P.O. Box 224
Monroe, CT 06468
Telephone: (203) 261-1234
Facsimile: (203) 261-5676
USPTO Customer No. 004955

A Very Low Bit Rate Speech Coder Based on a Recognition/Synthesis Paradigm

Ki-Seung Lee, *Member, IEEE,* and Richard V. Cox, *Fellow, IEEE*

Abstract—Recent studies have shown that a concatenative speech synthesis system with a large database produces more natural sounding speech. We apply this paradigm to the design of improved very low bit rate speech coders (sub 1000 b/s). The proposed speech coder consists of unit selection, prosody coding, prosody modification and waveform concatenation. The encoder selects the best unit sequence from a large database and compresses the prosody information. The transmitted parameters include unit indices and the prosody information. To increase naturalness as well as intelligibility, two costs are considered in the unit selection process: an acoustic target cost and a concatenation cost. A rate-distortion-based piecewise linear approximation is proposed to compress the pitch contour. The decoder concatenates the set of units, and then synthesizes the resultant sequence of speech frames using the Harmonic+Noise Model (HNM) scheme. Before concatenating units, prosody modification which includes pitch shifting and gain modification is applied to match those of the input speech. With single speaker stimuli, a comparison category rating (CCR) test shows that the performance of the proposed coder is close to that of the 2400-b/s MELP coder at an average bit rate of about 800-b/s during talk spurts.

Index Terms—Concatenative speech synthesis, piecewise linear approximation, rate distortion theory, very low bit rate speech coding.

I. Introduction

CONTEMPORARY speech coders such as CELP, MELP, MBE, or WI provide good quality speech at bit rates as low as 2400 b/s. However, for very low bit rates on the order of 100 b/s, these coders are unable to produce high quality speech, due to the reduced number of bits available for accurate modeling of the signal. In an effort to overcome this limitation, a new speech coder is proposed. This coder employs a different paradigm than conventional speech coders and is meant for applications where there are no delay or complexity limitations. For example, such a coder is very useful when requiring storage of large amount of pre-recorded speech. A talking book [4], which is a spoken equivalent of its printed version, requires huge space for storing speech waveforms unless a high compression coding scheme is applied. Similarly, for a wide variety of multimedia applications, such as language learning assistance, electronic dictionaries and encyclopedias there are potential applications of very low bit rate speech coders.

Techniques for a very low bit rate speech coder are based on what has been learned from previous work in speech coding, text-to-speech (TTS) synthesis, and speech recognition. Several groups of researchers have worked on a TTS-based approach. In TTS, synthesized speech can be produced by concatenating the waveforms of units selected from a large database. Prosody modification is often included as a post-processor for TTS systems. This typically adjusts the time scale and/or pitch to modify the prosody. Thus, a TTS-based coding scheme can be thought of as a speech coder that has a very large codebook composed of raw speech signals with additional parameters for compensating prosodic difference between the synthesized and the original speech signal. The first study using this approach was performed by Gerard *et al.* [3]. In this work, a text message and spoken utterance are jointly used to provide a TTS input stream and a small number of prototype pitch patterns and duration patterns are used for prosody coding. Bradley [4] introduced a wide-band speech coder which uses TTS to generate synthetic speech from text and then uses speech conversion to convert voice characteristics including speaking style, and emotion. This coder operates at 300 b/s. However, both these two coders necessarily require text transcription.

A speech coding system based on automatic speech recognition and TTS synthesis, which employed hidden Markov model (HMM)-based phoneme recognition and pitch synchronous over lap addition (PSOLA)-based TTS was proposed by Chen *et al.* [6]. This coder is referred to as the "phonetic vocoder" where the individual segments are quantized using a phonetic inventory. The reported bit rate was 750 b/s and the reconstructed speech quality was above a mean opinion score (MOS) of 3.0. For all TTS-based coders, since a speech signal is produced by TTS, the quality is highly dependent on the performance of the underlying TTS.

Alternatively, a segmental vocoder is also proposed to achieve very low bit rate. This coder attempts to decompose a speech signal into a sequence of segments that are subsequently quantized using a codebook of pre-stored segments. A typical example of this type of coder is the waveform segment vocoder by Roucos *et al.* [20]. In this, segmentation was performed in a very simple way, detecting regions with large spectral time-derivation, then a template sequence is constructed by minimizing distortion between a time-normalized template and an input segment. Since each template is a waveform segment that contains an excitation component as well as a spectral envelope, this coder does not need to transmit excitation signals which generally require a lot of bits in conventional speech coders. The

Manuscript received October 12, 1999; revised March 9, 2001. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Peter Kabal.

K.-S. Lee was with Shannon Laboratories, AT&T Laboratories—Research, Florham Park, NJ 07932-0971 USA. He is now with the Human & Computer Interaction Laboratory, Samsung Advanced Institute of Technology (SAIT), Ki-Heung, Korea (e-mail: ksleel@sait.samsung.co.kr).

R. V. Cox is with Shannon Laboratories, AT&T Laboratories—Research, Florham Park, NJ 07932-0971 USA.

Publisher Item Identifier S 1063-6676(01)04976-8.

bit rate of this coder is about 300 b/s. They obtained significantly less buzzy quality than their previous segmental coder, but, there were still artifacts in the coded speech signal, such as a "choppy quality," which mainly comes from the simple segmentation method. One of the limitations of this coder is the size of the template table, which is 9000. This number is not sufficient for representing the variability of the segments even though prosody modification is exploited to compensate for the difference between a template and an input segment.

A segmental vocoder using HMM-segmentation was proposed recently [1] in which template tables are constructed by a series of procedures, temporal decomposition, vector quantization and multigram segmentation. Each template segment is represented by a HMM. This approach is similar to the HMM-based phoneme recognition, but nonsupervised training was applied, thus the resultant segments do not correspond to phonetic inventories. This work mainly focused on the encoder part. Manipulation of prosody information was not discussed.

Although all of these methods are successfully applied to give extremely low bit rates, the common problem is that the quality of these coders is not satisfactory compared to conventional low rate coders ($\geq$1000 b/s) even when coding strategy focuses on a single speaker's voice. The quality of these coders is often not consistent and intelligibility is very bad at times. There are several reasons for this, including the relatively few templates in a typical system, the distortion introduced by using a speech representation that does not code speech transparently, audible discontinuities introduced by concatenation at segment boundaries, and the artifacts introduced by time scale modification.

The main goal of our work is to develop a speech coder whose quality is comparable to a conventional low rate speech coder (for example, a standard coding scheme at 2400 b/s), while maintaining bit rates lower than 1000 b/s. The basic idea is motivated by waveform-concatenation TTS systems, where a speech signal is produced by concatenating a selected unit sequence [15]. We utilize a large TTS labeled database as the "codebook" for our speech coder. The codebook contains several hours of speech, typically filled with phonetically balanced sentences. The identities of the phonemes, their durations, their pitch contours and all speech coding parameters are included in the database. Our approach to unit selection is different in that we use a frame as the basic synthesis unit and introduce a concatenation cost in order to reduce the distortions between neighboring units. This frame-based approach has the advantage that we can accurately choose units with short unit length. In addition, since frame selection does not require a time-warping process, we can synthesize the speech signal without time scale modification. The bit rate of a frame-based approach will be greater, because longer segments contribute to the high compression ratio of segmental coders. To cope with this problem, we design a selection process that increases the number of consecutive frames and subsequently apply run-length coding.

The remaining issue associated with a very low rate coder is the accurate coding of the pitch(F0) contour. This plays an important role in a very low rate coder since the correct pitch contour will increase naturalness and an efficient coding scheme will provide high coding gain. Nevertheless, most of the previous very low rate coders neglect this important issue. A possible way to reduce the number of bits in the pitch contour coding is to use schemes relying on a parametric description of the pitch contour [7]–[13]. In a parametric model, a segmental pitch contour is represented by a function and appropriate variables. The resulting information for representing the pitch contour is very small. Studies in this direction have been performed in applications requiring simpler representation of the pitch contour, such as intonation pattern analysis [8], [9], [11], [12] and automatic generation of the pitch contour in TTS systems [7], [13]. However, fundamental issues for application to a coding paradigm, such as the number of bits for representing model parameters and quantitative analysis of model error according to bit allocation have not been discussed.

The principle of our coding scheme is piecewise linear approximation of pitch that replaces the original pitch contour by consecutive lines. Techniques that minimize overall bit rate while maintaining approximation error below a given threshold will be described in more detail in Section V.

This paper is organized as follows. Section II gives an overview of our coder. Section III then describes the unit selection algorithm. The compression method of the unit sequence is presented in Section IV. In Section V we describe an efficient pitch contour coding method. Section VI presents the experimental results obtained from single speaker's corpus. We then conclude in Section VII with a discussion of the significant results and possible extensions.

II. Overview of the Coder

In a unit selection-based waveform-concatenating TTS scheme, synthesized speech is produced by concatenating the waveforms of units selected from a large database [15]. Units are selected to produce natural sounding speech of a given phoneme sequence predicted from text. This scheme has been widely used in several current TTS systems and gives synthetic speech that is close to natural. At this point, it can be assumed that if we replace parameters from text with those from a given speech signal, the resulting speech signal from TTS would sound like the input speech signal. This scenario, which is the basic scheme of the proposed speech coder, is depicted in more detail in Fig. 1.

We use mel-frequency cepstrum coefficients (MFCCs) as feature parameters for the unit selection. MFCCs have been widely used in both automatic speech recognition and speaker identification tasks. MFCCs as a unit selection parameter can provide reasonable intelligibility. In computing mel-cepstrum coefficients, a Hanning window of 25 ms at a frame rate of 100 Hz is applied. This means the length of each unit is 10 ms. In [19], the inclusion of features relevant to prosody increased naturalness of the synthesized signal, because decreased prosodic modification tends to reduce the artifacts of the synthesized speech. However, our experiment showed that introducing prosodic feature to the selection criteria sometimes produced lower intelligibility than the MFCCs-only case.

There are two databases in Fig. 1. The first one is for the unit selection process that contains MFCCs obtained in the same way as the feature extraction. The second one contains speech waveforms or appropriate coding parameters that are used to

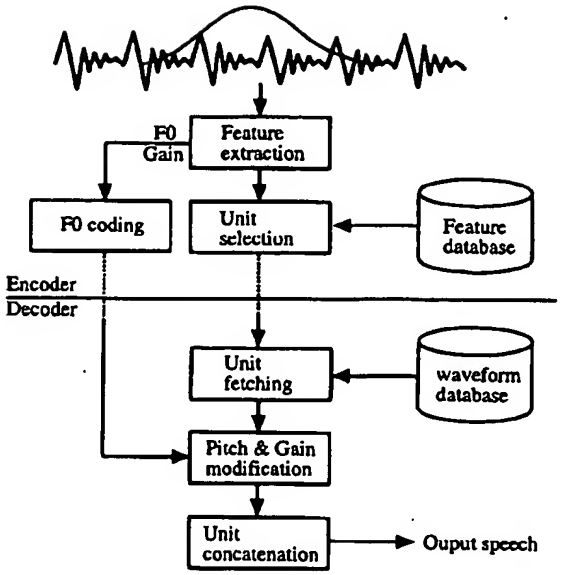


Fig. 1. Block diagram of the proposed coder.

make the output waveform. The raw speech signal from which MFCCs of the first database are computed is the same as those in the second database. Transmitted information, as shown in Fig. 1 are $F0$, gain, and unit indices. A unit index actually represents the position where the selected unit is located in the database.

A primary difference from the conventional coder is that we do not use any speech generation framework such as a source-filter model. We assume that any speech signal can be reconstructed by concatenating pitch modified short-segment waveforms that are adequately chosen from the large database. Another difference is that since the output speech is produced from a separate waveform database, the sampling rate of output speech is fully independent of the input speech sampling rate. That means, even if the input speech is a narrow band signal (i.e., $f_s = 8$ kHz), a wide band signal (i.e., $f_s = 16$ kHz) can be obtained. We used an $F0$ estimation method that is used in the waveform interpolation coder [16].

III. Unit Selection

In this paper, the problem of unit selection is formulated as how to find the optimal sequence from a large database in the sense of minimizing distortion within an individual frame and preserving natural coarticulation. That means there should be two cost functions in the unit selection process, target cost within an individual frame and concatenation cost between frames. When synthesizing a speech signal with an input feature vector sequence $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ by one from the synthesis database $\mathbf{U} = \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_T$, the total cost, $C(\mathbf{X}, \mathbf{U})$, is defined by summing the acoustic target cost, $C_A(\mathbf{x}_t, \mathbf{u}_t)$, and the concatenation cost, $C_C(\mathbf{u}_{t-1}, \mathbf{u}_t)$

$$C(\mathbf{X}, \mathbf{U}) = \sum_{t=1}^{T} w(\mathbf{x}, \mathbf{u}_t) C_A(\mathbf{x}_t, \mathbf{u}_t) + \sum_{t=2}^{T} C_C(\mathbf{u}_{t-1}, \mathbf{u}_t) \tag{1}$$

where

$$C_A(\mathbf{x}_t, \mathbf{u}_t) = \sum_{i=1}^{n} \left(c_{x_{t,i}} - c_{u_{t,i}}\right)^2 \tag{2}$$

$$C_C(\mathbf{u}_{t-1}, \mathbf{u}_t) = \sum_{i=1}^{n} \left(c_{u_{t-1,i}} - c_{u_{t,i}}\right)^2 \tag{3}$$

where $c_{x_{t,i}}$ and $c_{u_{t,i}}$ represent the i-th MFCC of the tth input frame and the unit $\mathbf{u}_t$, respectively, and n is the order of the MFCC. In (1), $w(\mathbf{x}_t, \mathbf{u}_t)$ represents fundamental frequency ($F0$) penalty at time t which increases the cost of selecting units with different $F0$s than the input. A possible penalty is given by

$$w(\mathbf{x}_t, \mathbf{u}_t) = \alpha \left(1 + \left|\log\left(\frac{F0_{x_t}}{F0_{u_t}}\right)\right|\right) \tag{4}$$

where $F0_{x_t}$ and $F0_{u_t}$ represent the fundamental frequency of the t-th input frame and the unit $\mathbf{u}_t$, respectively. There is a special condition for the concatenation cost. $C_C(\mathbf{u}_{t-1}, \mathbf{u}_t)$ is defined to be zero, if $\mathbf{u}_{t-1}$ and $\mathbf{u}_t$ are consecutive in the database. This encourages the selection of consecutive frames in the database which have natural coarticulation. In (3), the amount of unnaturalness between neighboring frames is assumed to be the Euclidean distance between their MFCCs. This is a reasonable assumption because a smoothly evolving spectral envelope over time increases the reconstructed speech quality [21]. Further improvement can be obtained by introducing an auditory-based distance measure with application to concatenative speech synthesis [22]. The optimal unit sequence $\mathbf{U}^*$ is obtained by minimizing the total cost $C(\mathbf{X}, \mathbf{U})$

$$\mathbf{U}^* = \arg \min_{\mathbf{U} \in \mathbf{U}_T} C(\mathbf{X}, \mathbf{U}) \tag{5}$$

where $\mathbf{U}_T$ is the set of all possible sequences that have T-units. This minimization can be performed by a Viterbi search processing one input unit at a time. Let $\mathbf{u}_t(i)$ be the ith unit at time t, the forward recursion is as follows:

$$\begin{aligned} C_t(i) &= \min_{1 \le j \le N} \{C_{t-1}(j) + C_C(\mathbf{u}_{t-1}(j), \mathbf{u}_t(i))\} \\ &\quad + C_A(\mathbf{x}_t, \mathbf{u}_t(i)) \\ \Psi_t(i) &= \arg \min_{1 \le j \le N} \{C_{t-1}(j) + C_C(\mathbf{u}_{t-1}(j), \mathbf{u}_t(i))\} \end{aligned} \tag{6}$$

where $1 \le i \le N$, $1 \le t \le T$, and $\Psi_t(i)$ is the backtracking pointer for the ith unit at time t, $C_t(i)$ is the accumulated cost for the ith unit at time t, and $C_C(\mathbf{u}_{t-1}(j), \mathbf{u}_t(i)) = 0$ if $\mathbf{u}_{t-1}(j), \mathbf{u}_t(i)$ are consecutive in a database.

After the final accumulated costs $C_T(i)$ for all i have been computed, the best unit sequence $\mathbf{U}^* = \mathbf{u}_1(q_1^*), \mathbf{u}_2(q_2^*), \ldots, \mathbf{u}_T(q_T^*)$ is obtained using the following backward recursion:

$$\begin{aligned} q_T^* &= \arg \min_{1 \le i \le N} C_T(i) \\ q_t^* &= \Psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1. \end{aligned} \tag{7}$$

Another criterion is to find the best sequence in the sense of maximizing the number of consecutive frames as well as minimizing the cost function. Before computing costs in (6), we find the maximum accumulated number of consecutive frames and which paths have this maximum number. Finding a minimum cost path is then performed by the above forward recursion. This can be implemented by introducing an accumulated number of

consecutive frames, $L_t(i)$ for the ith candidate at time t to the Viterbi decoding. The modified forward recursion is as shown in (8) at the bottom of the page, where $1 \le i \le N, 1 \le t \le T$, and $l_t(i,j)$ is the accumulated number of the consecutive frames up to unit $u_t(i)$ and a set $A_t(i)$ contains previous unit indices which have the maximum consecutive frames up to unit $u_t(i)$. This significantly improves the performance of the coder in quality and bit rates, since longer consecutive frames preserve natural coarticulation speech, and the efficiency of the subsequent run-length coding is increased when the number of consecutive frames is long.

In the above equations, any unit is assumed to be chosen from N units in the database. Because of the large size of the database (in this work, about 460 000, so $N = 460\,000$) the Viterbi search must be pruned to reduce the computational time. A pruning strategy will be described in the following sections.

A. VQ-Based Candidate Selection

In TTS, the number of possible units at a time is limited by the phoneme identification. A similar approach is employed in this paper, we focus on the limited number of units whose spectral envelope is relatively close to that of input frame. Since a set of the units close to the input frame occupies only a small portion of the entire database space, this can significantly reduce the computational complexity. This process requires partitioning the entire database space. We used vector quantization (VQ) for clustering. Supposing that a given unit is vector quantized by a specific code vector, the units quantized into the same code vector in a database are selected as candidate frames. If each frame corresponds to a phonetic inventory, the codebook size is six bits, or 64, which is nearest the number of phonemes, 51. An experiment showed that this number is too small to represent the variability of the frames and results in poor performance. Experimentally, we obtained good results when the codebook size was 10 or 11 bits.

This simple method has a problem due to the hard-clustering property of VQ [21]. As described in Fig. 2, when an input unit is close to a border of the space partitioning, more adequate candidate units may not be selected. To alleviate this, it is necessary to choose more than one cell. Well-known soft clustering techniques, such as Gaussian mixture model (GMM) or fuzzy clustering can be considered to choose the multiple cells. In our method, a relative distance measure is used. The Euclidean distance between the input and the VQ centroid is computed and reordered. Then, the ith cell is selected if the ratio $d_i/d_{\min}$ is greater than a given threshold (typically 0.7), where d_i is the distance between the input and the centroid of the ith cell and $d_{\min}$ is the minimum of d_i.

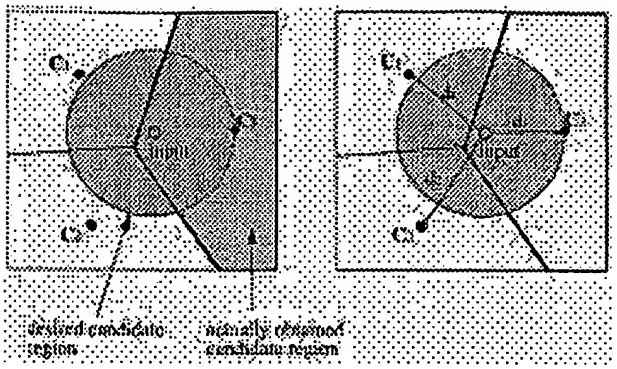


Fig. 2. VQ-based candidate selection. Three cells, C_1, C_2, and C_3, are the candidate cells in this example.

The number of candidate units depends on the number of candidate cells. In order to keep computational complexity from increasing greatly, we limit the number of candidate cells to six. The final procedure for selecting candidate units is to pick units within a hyper sphere with the input vector as a center. The radius determines the maximum allowable error of the candidate selection, which is closely related to the number of candidates. We determine the maximum allowable error by the bisectional search algorithm which is known as a fast search algorithm. The algorithm finds the maximum allowable error iteratively until the number of candidates reaches the desired number, N_d. Let us describe the algorithm for taking N_d candidates within a threshold Thres.

1) Compute the acoustic target costs, $C_A(x_i, u_i)$ of all the candidates within a candidate region.
2) Set initial Thres, N_d and $\Delta\text{Thres} = 0.5 * \text{Thres}$.
3) Count N = The number of candidates whose $C_A(x_i, u_i) \le \text{Thres}$.
4) If $N \le N_d$ at the first iteration or $|N - N_d| < \delta$, stop iteration.
5) If $N > N_d$, Thres = Thres − ΔThres, otherwise, Thres = Thres + ΔThres.
6) $\Delta\text{Thres} = 0.5 * \Delta\text{Thres}$, go to step 3.

Note that since ΔThres decrease exponentially, a small number of iterations is required in this procedure. This means that the above method is much faster than a full sorting-based selection method.

$$
\begin{aligned}
l_t(i,j) &= L_{t-1}(j) + \begin{cases} 1, & \text{if } u_{t-1}(j), u_t(i) \text{ are consecutive in a database} \\ 0, & \text{otherwise} \end{cases} \qquad 1 \le j \le N \\
l_t^*(i) &= \max_{1 \le j \le N} l_t(i,j) \\
A_t(i) &= \{k \mid l_t(i,k) = l_t^*(i), k = 1, \ldots, N\} \\
\tilde{\Psi}_t(i) &= \arg\min_{j \in A_t} \{C_{t-1}(j) + C_C(u_{t-1}(j), u_t(i))\} \\
L_t(i) &= l_t(i, \tilde{\Psi}(i)) \\
C_t(i) &= \min_{j \in A_t} \{C_{t-1}(j) + C_C(u_{t-1}(j), u_t(i))\} + C_A(x_t, u_t(i))
\end{aligned} \tag{8}
$$

B. Context-Based Viterbi Pruning

The typical number of candidates after VQ-based candidate selection ranges from 500 to 1000. This number corresponds to only 0.1% or 0.2% of the entire units in the database. However, the Viterbi decoding process still requires lots of computation.

The pruning strategy of this section is based on a contextually meaningful criterion. Since the proposed speech coder uses a TTS database, which is already phonetically labeled, we can predict whether a given path is possible in some context. For example, assuming that a database is labeled using half phones, the following frame of the current frame labeled with "aa1" (aa1 is the first half of phoneme aa) must have phoneme "aa1" or "aa2" (aa2 is the last half of phoneme aa). All other combinations, like aa1-ae1 or aa1-k2, must be removed. In this way, we can reduce the number of paths in the Viterbi algorithm. Experimental results showed that approximately 50% of the total number of paths have contextually legal combination of phonemes. This means by using this pruning, the amount of computation for concatenation cost can be reduced by 50%. The sound quality after this pruning process was almost the same or somewhat better than that from the method without pruning. In terms of computational complexity and size of the memory, this pruning process requires just one character comparison and no additional memory.

IV. Coding the Selected Unit Sequence

Since the concatenation cost in (1) is set to zero if two frames are consecutive in a database, the resulting unit sequence has many consecutive frames. To take advantage of this property, a run-length coding technique is employed to compress the unit sequence. In this method, a series of consecutive frames are represented with the start frame index and the number of the following consecutive frames as shown in Fig. 3. Thereby a number of consecutive frames are encoded into only two variables.

The coding efficiency of the example in Fig. 3 is $(437 - 198)/(437)*100 = 54.7\%$. In this example, we assigned 19 bits for start frame index because of $\log_2 460\text{ K} \cong 19$ (bits). However, the possible units are limited by the phone index of the previous frame, as described in Section III-B, the actual number of possible units is less than the total number of units. The number of bits for a start frame index is determined according to the phone identification of the last frame. For example, if the last frame has a phone "aa1" and the number of occurrences of aa1 in a database is 10 K, the required bit for the following frame index is $\log_2 10\text{ K} \cong 14$ (bits), instead of 19 bits.

As for the bits for quantizing a length of consecutive units, which is referred to as a run-length in this paper, a variable bit allocation proved to be more efficient than a fixed bit allocation. Experimental results showed that about 30 b/s were saved by using Huffman coding. This is ensured by the histogram of the run-length in Fig. 4. The corresponding Huffman code table is also shown in Fig. 4. As shown, the smaller number of bits are allocated for the shorter run-length.

V. Coding the $F0$ Contour

In order to get a high compression ratio, our $F0$ coding is contour-wise rather than frame-wise. Piecewise linear approximation (PLA) [12], as shown in Fig. 5, is used to implement our contour-wise $F0$ coding. PLA seems to be very favorable for high compression, because we need transmit only a small number of sampled points instead of all individual samples. Of course, the intervals between the sampled points must be transmitted for proper interpolation. In general, the total number of bits for PLA is smaller than frame-wise coding.

PLA always presumes some degree of smoothness for the function approximated. Therefore, we apply a median smoothing filter to the $F0$ contour before compressing it. Gross representation of the $F0$ contour by piecewise linear approximation causes larger coding errors than frame-wise coding. This error depends on how to select $F0$ samples as endpoints of the approximation lines. Therefore, an optimizing PLA is formulated for finding the locations of $F0$ points by minimizing the error between the $F0$ contour and the approximation. Two methods for finding the location of $F0$ points are proposed in this work. In the following sections, we discuss these issues in more detail.

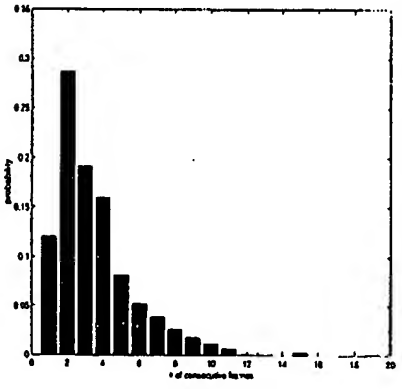

```
A unit sequence (Before coding)

189014  189015  189016  454609  154299  369469
369470  369471  417562  417563  417564  417565
417567  417568  417569  417570  417571  335566
335567  335568  368826  368827
 20204  (Total 19*23=437bits)

After run length coding

189014 3  454609 1  154299 1  369469 3  417562 4
417567 5  335566 3  368826 2   20204 1  (Total
22*9=198bits)
```

Fig. 3. Bit fields for a (top) consecutive unit sequence and (bottom) an example.



run length	code
1	000
2	10
3	01
4	110
5	1110
6	0010
7	11110
8	11111
9	00110
10	001110
11	0011110
12	00111111000
13	00111111
15	001111101
18	00111110011
Escape	00111110010

Fig. 4. (left) Histogram of run-length and (right) its Huffman code table.

A. Successive Linear Approximation

The method introduced in this section is close to the polygon approximation [18] algorithm applied in image coding applications. It was developed for efficient compression of two-dimensional (2-D) polygons. Successive approximation for $F0$

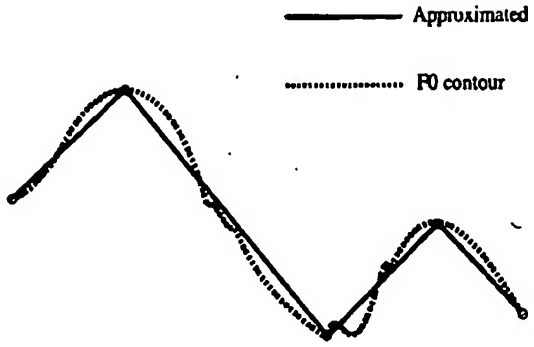


Fig. 5. Piecewise linearly approximation.

coding can be thought of as a one-dimensional (1-D) version of the polygon approximation.

Fig. 6 depicts the framework of the successive linear approximation for the $F0$ contour. Linear approximation is carried out using those two contour points with the maximum error between them as the starting point. Then, additional points are added to the line where the error between the approximated and contour are maximum. This is repeated until the contour approximation error is less than $d^*_{\max}$. The resulting approximated contour guarantees that the approximation error is below $d^*_{\max}$.

This method considers only instantaneous error. However, mean squared error is sometimes a more meaningful criterion than instantaneous error. Overlooking this measure causes larger mean squared error in some regions, even if a small $d^*_{\max}$ is met. To alleviate this problem, we modified the successive approximation mentioned above to achieve better performance. In the modified method, the approximation is carried out according to the following steps.

1) Compute the mean square error for each line, and find the line with maximum mean squared error among all approximation lines.
2) For the line with maximum mean squared error, pick the point with maximum error between the original contour and the approximated line.
3) If the maximum error is greater than $d^*_{\max}$, go to Step 4), otherwise stop approximation.
4) Add point from Step 2) to the line from Step 1), and go to Step (1).

Note that the mean squared error criterion is used for preselection of the $F0$ point for linear approximation. This leads to regions with high fluctuation that are subsequently piecewise linearly approximated. According to experiments, even with the same $d^*_{\max}$ threshold, the mean squared error over the whole $F0$ contour is further reduced by the modified algorithm.

Determining the threshold error $d^*_{\max}$ is extremely crucial, as this value affects both the number of bits and the perceptual quality. During subjective evaluations of synthesized speech signals, it was found that allowing a maximum error of $d^*_{\max} = 5$ Hz for a female talker is sufficient to allow proper representation of the $F0$ contour as well as obtaining a reasonable bit rate.

The B-spline approximation for $F0$ contour was also considered in this work. Visual inspection revealed that the approximated contour by B-spline was closer to the original $F0$ contour due to its smoother representation of the contour. However there was no clear perceptual difference. Hence, we concluded that linear approximation is good enough for representing $F0$ contour.

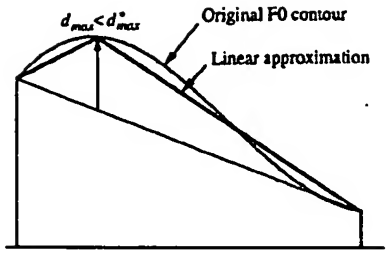


Fig. 6. Successive linear approximation for $F0$ contour.

B. Linear Approximation Based on Rate-Distortion Criterion

In this section, we propose an optimal method that takes into account not only the approximation error but also the number of bits. The method is implemented based on rate distortion criteria.

Let $\mathbf{P} = \{p_0, \ldots, p_{N_p-1}\}$ denote the set of $F0$ points used to approximate the contour, which is also an ordered set, with N_p, the total number of $F0$ points in $\mathbf{P}$, and the k-th line starting at p_{k-1} and ending at p_k. Since $\mathbf{P}$ is an ordered set, the ordering rule and the set of points uniquely define the approximated contour.

Now, we define a constrained minimization problem

$$\text{Minimize } R(\mathbf{P}) \quad \text{subject to } D_{\max}(\mathbf{P}) \le d^*_{\max} \tag{9}$$

where $R(\mathbf{P})$ is the total number of bits needed to encode the $F0$ set $\mathbf{P}$ including values and positions, and $D_{\max}(\mathbf{P})$ is the overall maximum absolute error defined by

$$D_{\max}(\mathbf{P}) = \max_{k \in \{1,\ldots,N_p-1\}} d_{\max}(p_{k-1}, p_k) \tag{10}$$

where $d_{\max}(p_{k-1}, p_k)$ is the maximum absolute error between the line p_{k-1} to p_k and actual $F0$ values. Note that there is an inherent tradeoff between $R(\mathbf{P})$ and $D_{\max}(\mathbf{P})$ in the sense that a small $D_{\max}(\mathbf{P})$ requires a high $R(\mathbf{P})$, whereas a small $R(\mathbf{P})$ results in a high $D_{\max}(\mathbf{P})$.

To find an easier way to solve the problem, we rewrite $R(\mathbf{P})$ in (6) as follows:

$$R(\mathbf{P}) = \sum_{k=0}^{N_p-1} \omega(p_{k-1}, p_k) \tag{11}$$

where

$$\omega(p_{k-1}, p_k) = \begin{cases} \infty, & \text{if } d_{\max}(p_{k-1}, p_k) \ge d^*_{\max} \\ r(p_{k-1}, p_k), & \text{otherwise} \end{cases} \tag{12}$$

where $r(p_{k-1}, p_k)$ is the number of bits needed to encode line p_{k-1} to p_k.

Now, the problem can be formulated in the form of a directed graph, as shown in Fig. 7. The vertices of the graph correspond to the admissible $F0$ points, and edges correspond to the possible segments of the approximation line. The edges have weights $\omega(p_{k-1}, p_k)$. The total number of bits $R(\mathbf{P})$ is propor-

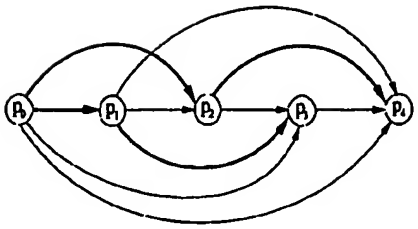


Fig. 7. Example of directed graph for the linear approximation. The bold line means local minimal path.

tional to the number of points N_p. Thus the problem can be considered as a problem of finding a shortest path. Note that the above definition of the weight function leads to a length of infinity for every path that includes a line segment resulting in an approximation error larger than $d^*_{\max}$. We can find an optimal path by exhaustive search for all possible set $\mathbf{P} = \{p_0, \ldots, p_{N_p-1}\}$. However, this is not a practical way because of the quite expansive computational cost. As an alternative, dynamic programming is employed. It first finds a "*local minimal path*" for all $F0$ points within a syllabic contour, then the global minimum path is built by backtracking. The overall procedure for finding an optimal set of $F0$ points $\mathbf{P}^* = \{p^*_0, \ldots, p^*_{N_p-1}\}$ is as follows:

$$w(n) = \arg \min_{1 \le k < n} \{\alpha_{k,n}[R(p_k) + r(p_k, p_n)]\}$$
$$R(p_n) = \{R(p_{w(n)}) + r(p_{w(n)}, p_n)\}$$
$$D_{\max}(p_n) = \max\{D_{\max}(p_{w(n)}), d_{\max}(p_{w(n)}, p_n)\} \quad (13)$$

where $1 \le n \le N-1$ and N is the total number of $F0$ samples within a syllabic contour, $R(p_k)$ is the accumulated number of bits up to p_k, similarly, $D_{\max}(p_k)$ is the maximum error up to p_k, and $\alpha_{k,n}$ is given by

$$\alpha_{k,n} = \begin{cases} \infty, & \text{if } \max\{D_{\max}(p_k), d_{\max}(p_k, p_n)\} > d^*_{\max} \\ 1, & \text{otherwise.} \end{cases}$$

The backtracking pointer, $w(n)$ holds an indication of which $F0$ point is the start point of the path with the minimum accumulated number of bits at p_n. The optimal sequence of $F0$ points in reverse order is

$$p^*_N, p^*_{w(N)}, p^*_{w(w(N))}, \ldots.$$

An example is given in Fig. 7. For each point p_n, the bold line denotes the local minimal path ($=w(n)$). It can be easily understood that the optimal set of $F0$ points after backtracking is given by $\mathbf{P}^* = \{p^*_0, p^*_2, p^*_4\}$.

VI. Experiments and Results

This section presents the experimental results of the proposed coder for a single female speaker. The size of the database in our work is about 76 min which corresponds to 460 K units. For the test, we also prepared 15 test sentences from the same talker.[1] All speech signals were recorded at 48 kHz in a noise-free environment, and low pass filtered to 7 kHz, then down-sampled at 16 kHz. Twenty-one MFCCs including the zeroth coefficient were computed for unit selection. A pre-emphasis factor of 0.95 is applied, and the number of mel-frequency filter banks are 24.

First, we evaluate the performance of the proposed $F0$ coding method. Fig. 8 shows the original $F0$ contour versus approximated contours for $d^*_{\max} = 5$ Hz and $d^*_{\max} = 10$ Hz, respectively. The results in the figure were obtained from rate-distortion criterion presented in Section V-B. A more coarse representation of a given $F0$ contour is found at higher $d^*_{\max}$ value. Practically, setting a maximum allowable error $d^*_{\max}$ to 5–6 Hz results in a perceptually good approximation for a female voice's $F0$ contour. We also encoded a number of $F0$ contours from the 15 sentences and averaged the resulting bit rates. The bit allocation for $F0$ information is summarized in Table I. The experiments were performed for various $d^*_{\max} = 1, 2, \ldots, 11$. The results are shown in Fig. 9. The shape of the resulting curve comes up with a general rate-distortion curve even though there is no explicit relationship between bit rate and $d^*_{\max}$. For the successive approximation case in Section V-A, results are almost the same as for the rate-distortion criterion, but the bit rate is slightly increased (135.9 b/s for the successive approximation method and 120.5 b/s for the rate-distortion-based method).

The average bit rate for each parameter is summarized in Table II. This result is also based on the 15 test sentences and the bit rate for $F0$ is from the method based on rate distortion with $d^*_{\max} = 6$ Hz. The bits for gain information were determined according to the method described in [24], however this method originally required phonetic segmentation which is not available in this work. Hence, we used a simple segmentation method which is based on the voiced/unvoiced decision and the first-order orthogonal polynomial coefficients for MFCCs. The threshold for detecting segment boundaries was determined in a heuristic way which produces the same number as the phoneme boundaries. In the unit selection process, the modified forward recursion (8) and all the pruning methods described in Section III were used. As shown in Table II, more than 60% of the total b/s is occupied by the frame index. This is because the large size of the database entails more bits. The subjective listing test according to the size of the database will give useful clues to help decrease bit rates.

There are several ways to synthesize speech waveforms from the selected unit sequence, such as PSOLA [25], HNM [14], and MBROLA [26]. Among them, HNM-based synthesis gives good performance for prosody modification as well as concatenation, due to its parametric modeling approach. Hence, we adopted it for waveform synthesis. Since the HNM synthesis is pitch synchronous, there is time misalignment between the selected frame unit sequence and the HNM parameter sequence. Indeed, the female voice has a generally higher pitch and this leads to insufficient frame information when frames represent 10-ms intervals. Copying or deleting HNM parameters may be a solution for this problem. However, this causes annoying discontinuities and buzziness of the synthesized speech signal. In order to minimize quality loss at the synthesis stage, we employed a multimodal interpolation technique that applies different kinds of interpolation methods according to the characteristics of frame joining points. For example, if two frames are not naturally concatenated (in other words, the frame indices of the two frames are not consecutive) the HNM parameters of the

[1]The database does not include these test speech signals.

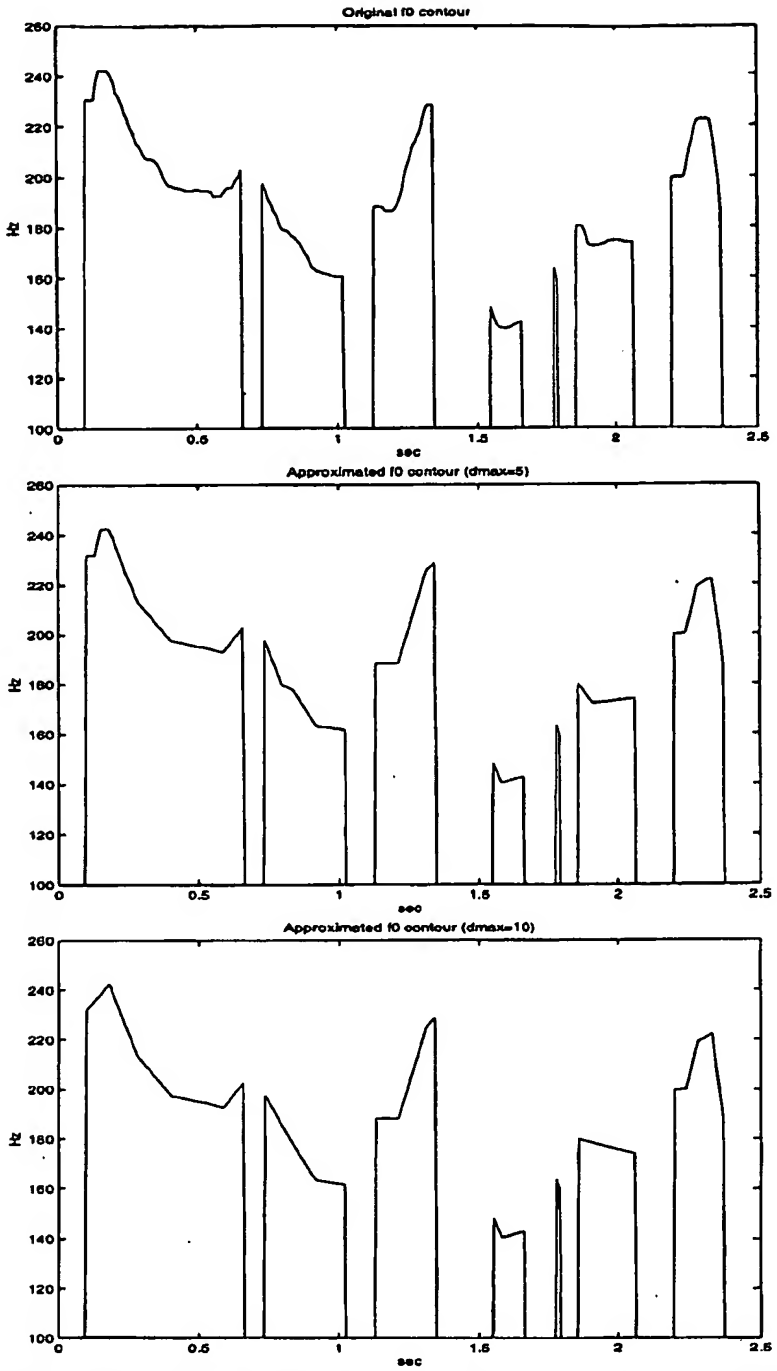


Fig. 8. (Top) Original $F0$ contour, (middle) approximated contour ($d^*_{\max} = 5$ Hz), and (bottom) approximated contour ($d^*_{\max} = 10$ Hz).

TABLE I
BIT ALLOCATION OF $F0$ INFORMATION

PARAMETERS	ALLOCATED BITS
F0 (VALUE)	8
F0 (DURATION)	5
GAIN	5

intermediate frames are obtained by the interpolation of those of neighboring frames. It is well known that a high degree of discontinuity can be expected when the speech signal changes from unvoiced to voiced and vice-versa. In other words, preserving the discontinuities at the voicing status changing points provides more natural sounding speech. Nearest neighbor search is used to find the HNM parameters at joining points where the voicing states of the neighborhoods are different from each other. Note that MBROLA uses constant frame length at synthesis time, this feature will reduce the complexity of HNM synthesis.

A subjective formal listening test was conducted to compare speech quality of the unit selection-based waveform concate-

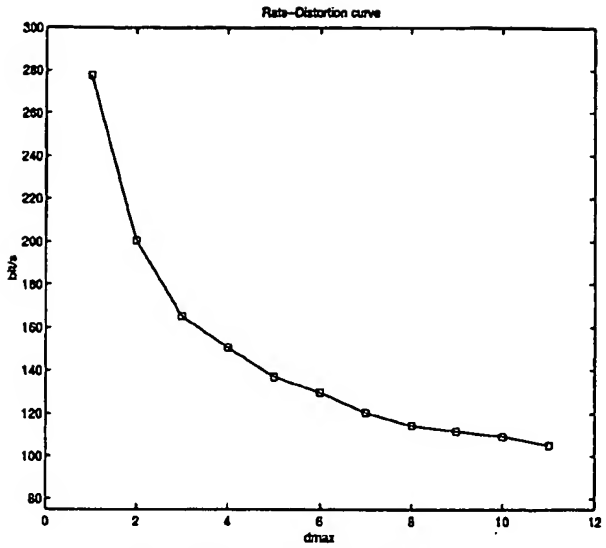


Fig. 9. Rate versus maximum distortion curve.

TABLE II
AVERAGE VALUES OF BIT RATE FOR EACH PARAMETER

PARAMETERS	BIT/S
FRAME INDEX	521.7
RUN LENGTH	91.7
F0	120.5
GAIN	99.5
TOTAL	833.4

nation and conventional speech coders. The modified forward recursion produced much better sound quality than the original forward recursion (6), thus we used the results from the modified method as test speech signals. Since the goal of this work is to produce synthetic speech whose quality is comparable to conventional low bit rate coders, overall user acceptability of the reconstructed speech has been measured with a comparison category rating (CCR) test [17]. The listeners identify the quality of the second stimulus relative to the first using a two sided rating scale, as shown in Table III. Thirteen listeners participated and were asked to judge which stimulus is better or worse than the other. Each stimulus consisted of the 8-kHz downsampled reconstructed speech from the proposed coder and the reconstructed speech from the 2400 b/s MELP coder [27]. The speech was from the test data set. The contents of the test sentences are listed in Table IV. There are five different contents. Each sentence was uttered three times with three different prosodies. Thus, total $3 * 5 = 15$ stimuli were evaluated by each listener. The average CCR was -0.28, the maximum CCR was 0.33 and the minimum CCR was -0.64. For all five sentences, CCRs are less than 1. This means the quality of the proposed coder is close to that of the 2400 b/s MELP coder. The listeners indicated that the distortions caused by the two speech coders sound different from each other. This is due to the fundamentally different approaches between the two coders. The major factors of quality degradation of the proposed coder are large distortion between the input cepstrum and the one from the selected unit, pitch modification and interpolation of HNM parameters.

TABLE III
QUALITY RATING SCALE FOR A CCR TEST

DESCRIPTION	RATING
MUCH BETTER	2
BETTER	1
ABOUT THE SAME	0
WORSE	-1
MUCH WORSE	-2

TABLE IV
CCR FOR EACH TEST SENTENCE

SENTENCE	RATING
"Two boyscouts stood watch outside."	-0.28
"Candy's purple gown looks awful."	-0.18
"I'm waiting for my pear tree to bear fruit."	-0.59
"We must complete every task."	-0.64
"He ate too much corn at the picnic."	0.33
Average	-0.28

Noisy or unclear qualities were sometimes found in unvoiced regions. Slight audible discontinuities were also found in the speech signal from the proposed coder though a concatenation cost is engaged in unit selection. These defects were more visible when comparing with the original 16-kHz sampled speech signals. However, according to CCR score, it appears that the overall quality of reconstructed speech signals is reasonable in both intelligibility and naturalness.

VII. CONCLUSION

A very low bit rate speech coder based on a new paradigm is proposed in this paper. The objective of this work is to make the quality of a speech coder operating at below 1000 b/s close to that of conventional low rate coders. The unit selection approach which has been widely used in TTS system is a key part of the encoder. An acoustic target cost function related to intelligibility and a concatenation cost related to naturalness are applied to unit selection. A technique which can provide longer consecutive frames is also introduced in order to increase sound quality as well as coding efficiency. Two pruning methods in a Viterbi decoder are introduced to reduce computation times. At the decoder, waveform concatenation and prosody modification are exploited to obtain the reconstructed speech signal. As a synthesis method, the HNM framework is used. Using MFCCs in unit selection was motivated by automatic speech recognition and speaker identification. As for $F0$ coding, we introduced linear approximation schemes in order to get an extremely low bit rate. A rate-distortion criterion is applied to the linear approximation. Using this criterion, we can implement an optimal method for minimizing bit rates with adjustable approximation error.

The experiment showed the effectiveness of the proposed schemes: prosodic information is preserved while $F0$ and gain undergo high compression. In a formal listening test, we confirmed that the quality of the proposed coder was very close to that of a conventional 2400-b/s coder.

This coder is limited to a single speaker's voice. If we limit the application to where only one speaker's voice is needed, such as a personalized communication system, the proposed coder can be successfully exploited. Otherwise, additional effort to achieve multiple speaker capability is needed. Increasing the size of the database to contain a number of speakers' utterances is a possible solution. Although work on voice personality conversion is still underway, a future voice personality transformation algorithm will be a solution for this multiple speaker capability.

REFERENCES

[1] J. Černocký, G. Baudoin, and G. Chollet, "Segmental vocoder-going beyond the phonetic approach," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 2, pp. 605–608, 1998.

[2] C. M. Ribeiro and I. M. Trancoso, "Phonetic vocoding with speaker adaptation," in *Proc. EUROSPEECH'97*, vol. 3, 1997, pp. 1291–1294.

[3] G. Benbassat and X. Delon, "Low bit rate speech coding by concatenation of sound units and prosody coding," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, pp. 1.2.1–1.2.4, 1984.

[4] P. Vepyek and A. B. Bradley, "Consideration of processing strategies for very-low-rate compression of wide band speech signal with known text transcription," in *Proc. EUROSPEECH'97*, vol. 3, 1997, pp. 1279–1282.

[5] M. Ismail and K. Ponting, "Between recognition and synthesis-300 bits/second speech coding," in *Proc. EUROSPEECH'97*, vol. 1, 1997, pp. 441–444.

[6] H. C. Chen, C. Y. Chen, K. M. Tsou, and O. T.-C. Chen, "A 0.75 Kbps speech codec using recognition and synthesis schemes," in *Proc. IEEE Workshop Speech Coding Telecommunications*, 1997, pp. 27–29.

[7] F. Malfrere and T. Dutoit, "High quality speech synthesis for phonetic speech segmentation," in *Proc. EUROSPEECH'97*, 1997, pp. 2631–2634.

[8] C. d'Alessangdro and P. Mertens, "Automatic pitch contour stylization using a model of tonal perception," *Comput., Speech, Lang.*, vol. 9, pp. 257–288, 1995.

[9] P. Taylor, "The rise/fall/connection model of intonation," *Speech Commun.*, vol. 15, no. 1/2, pp. 169–186, 1994.

[10] D. J. Hirst, P. Nicolas, and R. Espesser, "Coding the $F0$ of a continuous text in French: An experimental approach," in *Proc. Int. Congr. Phonetic Sciences*, 1991, pp. 234–237.

[11] H. Fujisaki and H. Kawai, "Realization of linguistic information in the voice fundamental frequency contour," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, pp. 663–666, 1988.

[12] M. T. M. Scheffers, "Automatic stylization of $F0$ contours," in *Proc. 7th FASE Symp.*, vol. 3, Edinburgh, U.K., 1988, pp. 981–984.

[13] J. Pierrehumbert, "Synthesizing intonation," *J. Acoust. Soc. Amer.*, vol. 70, no. 4, pp. 985–995, 1981.

[14] Y. Stylianou, T. Dutoit, and J. Schroeter, "Diphone concatenation using a harmonic plus noise model of speech," in *Proc. EUROSPEECH '97*, 1997, pp. 613–616.

[15] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, "The AT&T next-gen TTS system," in *Proc. Joint Meeting ASA, EAA, DAGA*, Berlin, Germany, Mar. 1999.

[16] W. B. Kleijn and J. Haagen, "Waveform interpolation for coding and synthesis," in *Speech Coding and Synthesis*, W. Kleijn and K. Paliwal, Eds. Amsterdam, The Netherlands: Elsevier, 1995, ch. 4, pp. 175–207.

[17] ——, "Evaluation of speech coders," in *Speech Coding and Synthesis*, W. Kleijn and K. Paliwal, Eds. Amsterdam, The Netherlands: Elsevier, 1995, ch. 4, pp. 467–493.

[18] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster, "MPEG-4 and rate-distortion-based shape-coding techniques," *Proc. IEEE, Special Issue Part Two: Multimedia Signal Processing*, vol. 86, no. 6, pp. 1126–1154, June 1998.

[19] A. J. Hunt and A. W. Black, "Concatenative speech synthesis using units selected from a large speech database,", Draft Paper.

[20] S. Roucos and A. M. Wilgus, "The waveform segment vocoder: A new approach for very-low-rate speech coding," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 236–239, 1985.

[21] H. P. Knagenhjelm and W. B. Kleijn, "Spectral dynamics is more important than spectral distortion," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 732–735, 1995.

[22] J. H. L. Hansen and D. T. Chappell, "An auditory-based distortion measure with application to concatenative speech synthesis," *IEEE Trans. Speech Audio Processing*, vol. 6, pp. 489–495, Sept. 1998.

[23] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall.

[24] K.-S. Lee and R. V. Cox, "TTS based very low bit rate speech coder," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 181–184, 1999.

[25] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Commun.*, vol. 9, no. 5/6, pp. 453–467, 1990.

[26] T. Dutoit and H. Leich, "Text-to-speech synthesis based on a MBE re-synthesis of the segments database," *Speech Commun.*, vol. 19, pp. 119–143, 1996.

[27] A. V. McCree and T. P. Barnwell, III, "A mixed excitation LPC vocoder model for low bit rate speech coding," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 242–250, July 1995.

Ki-Seung Lee (S'93–M'98) was born in Seoul, Korea, in 1968. He received the B.S., M.S., and Ph.D. degrees in electronics engineering from Yonsei University, Seoul, in 1991, 1993, and 1997, respectively.

From February 1997 to September 1997, he was with the Center for Signal Processing Research (CSPR), Yonsei University. From October 1997 to September 2000, he was with the Speech Processing Software and Technology Research Department, Shannon Laboratories, AT&T Laboratories—Research, Florham Park, NJ, where he worked on ASR/TTS-based very low bit rate speech coding and prosody generation of the AT&T NextGen TTS System. He is currently with the Human and Computer Interaction Laboratory, Samsung Advanced Institute of Technology (SAIT), Suwon, Korea. His research interests include the various fields of Text-to-Speech synthesis, image enhancement, speech coding, and general purpose DSP-based real-time implementation.

Richard V. Cox (S'69–M'70–SM'87–F'91) received the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ.

In 1979, he joined the Acoustics Research Department, Bell Laboratories, Murray Hill, NJ. He conducted research in the areas of speech coding, digital signal processing, analog voice privacy, audio coding, and real-time implementations. He is well-known for his work in speech coding standards. He collaborated on the low-delay CELP algorithm that became ITU-T Recommendation G.728 in 1992. He managed the ITU effort that resulted in the creation of ITU-T Recommendation G.723.1 in 1995. In 1987, he was promoted to Supervisor of the Digital Principles Research Group. In 1992, he was appointed Department Head of the Speech Coding Research Department, AT&T Bell Labs. In 1996, he joined AT&T Labs as Division Manager of the Speech Processing Software and Technology Research Department. In August 2000, he was appointed Speech and Image Processing Services Research Vice-President. In this capacity, he has responsibility for all of AT&T's research in speech, audio, image, video, and multimedia processing research. He is also Vice Chairman of the Board of Directors of Recording for the Blind and Dyslexic (RFB&D), the only U.S. provider of textbooks and reference books for people with print disabilities. At RFB&D, he is helping to lead the effort to develop digital books combining audio, text, images, and graphics. These "multimedia books" will be available in 2001 for RFB&D K-14 students throughout the U.S.

Dr. Cox is President-Elect of the IEEE Signal Processing Society. In 1999, he was awarded the AT&T Science and Technology Medal.